



汉钟绩效考核系统



目录

INDEX

01



系统架构

02



系统构思

03



系统逻辑

04



系统扩展

05



系统答疑



01



系统架构

Hanbell-KPI



Hanbell-KPI系统入口

部门指标

产品指标

员工指标

各项指标设定和详细数据

指标维护界面

指标更新排程

报表发送排程

各项指标，数据抽取、计算、归集接口

CRM DB

OA DB

ERP DB

MES DB

云端 DB

其他 DB



02



系统构思

Hanbell-KPI



考核
指标

指标名称 计算类型 计算接口...

目标值

Target

基准值

Benchmark

预测值

Forecast

实际值

Actual

绩效值

Performance

其他值

Other1-6



定义

为相同属性的指标定义一个模板
IndicatorDefine

分配

把模板或指标分配到具体考核部门
Indicator, IndicatorDetail

归集

把部分指标归集到某个相关指标
IndicatorSet



02



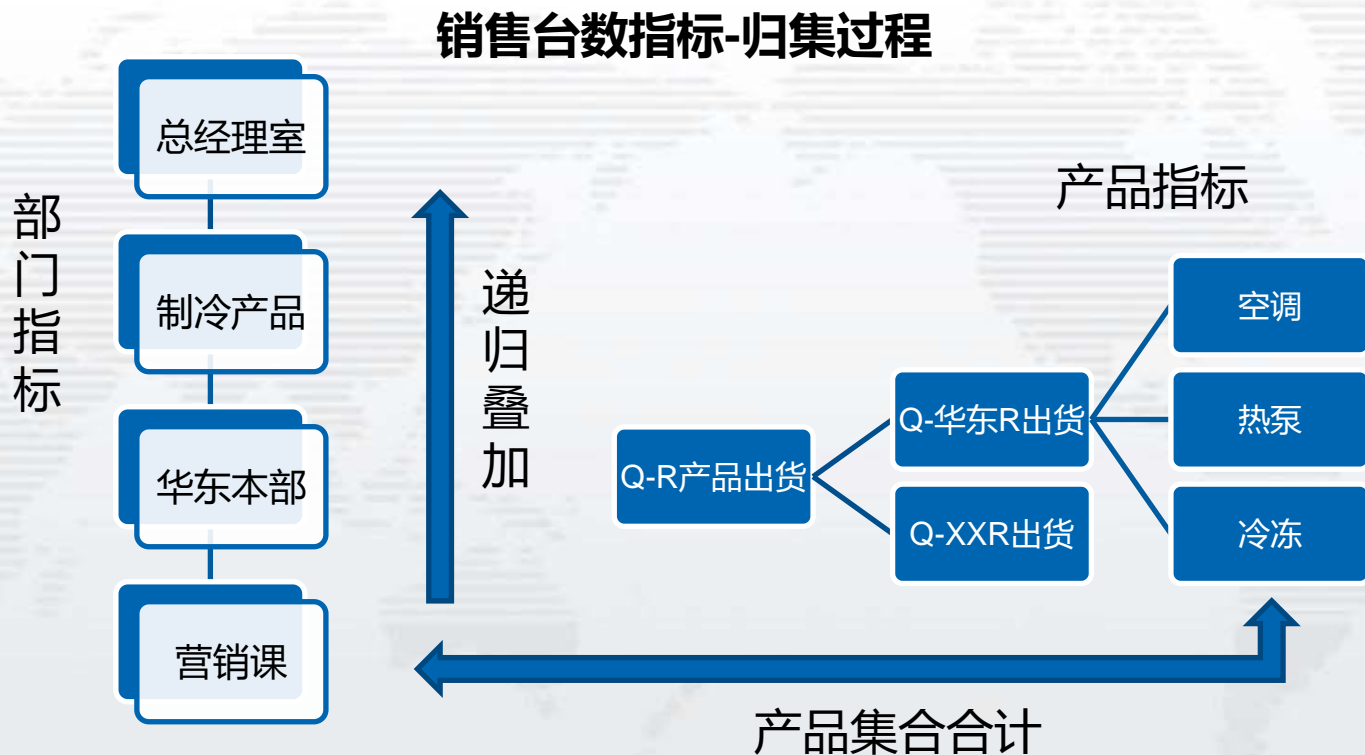
系统逻辑

Hanbell-KPI



销售台数指标-分配过程







接口

Interface

```
public interface Actual {  
  
    public void setEJB(String JNDIName) throws Exception;  
  
    public LinkedHashMap<String, Object> getQueryParams();  
  
    public int getUpdateMonth(int y, int m);  
  
    public int getUpdateYear(int y, int m);  
  
    public BigDecimal getValue(int y, int m, Date d, int type, LinkedHashMap<String, Object> map);  
  
}
```

抽象

Abstract

```
public abstract class ShipmentQuantity extends Shipment {  
  
    public ShipmentQuantity() {  
        super();  
    }  
  
    @Override  
    public BigDecimal getValue(int y, int m, Date d, int type, LinkedHashMap<String, Object> map) {...96 行 }  
  
}
```



实现

Implement

调用

Invoke

```
public class ShipmentQuantityR1B1 extends ShipmentQuantity {  
  
    public ShipmentQuantityR1B1() {  
        super();  
        queryParams.put("facno", "C");  
        queryParams.put("deptno", " '1B000' ");  
        //queryParams.put("decode", "1");  
        queryParams.put("n_code_DA", " = 'R' ");  
        queryParams.put("n_code_CD", " = 'HD' ");  
        queryParams.put("n_code_DC", " = 'R' ");  
        queryParams.put("n_code_DD", " = '00' ");  
    }  
  
}
```

```
if ((entity != null) && (entity.getSeq() == y) && (entity.getActualInterface() != null) && (!"".equals(entity.getActualInterface())) {  
    IndicatorDetail a = entity.getActualIndicator();  
    try {  
        actualInterface = (Actual) Class.forName(entity.getActualInterface()).newInstance();  
        uy = actualInterface.getUpdateYear(y, m);  
        um = actualInterface.getUpdateMonth(y, m);  
        actualInterface.setEJB(entity.getActualEJB());  
        BigDecimal na = actualInterface.getValue(uy, um, d, type, actualInterface.getQueryParams());  
        Method setMethod = a.getClass().getDeclaredMethod("set" + this.getIndicatorColumn("N", um).toUpperCase(), BigDecimal.class);  
        setMethod.invoke(a, na);  
        indicatorDetailBean.update(a);  
    } catch (Exception ex) {  
        Logger.getLogger(IndicatorBean.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```



谢谢观看

Thanks